# LAUNCHING AN INSTANCE OF RED HAT OPENSTACK PLATFORM WITH HORIZON (CL110)

Taste of Red Hat Training course excerpt

## TABLE OF CONTENTS

1 INTRODUCTION ................................................................................................................ 2

2 INDUSTRY OPPORTUNITY ............................................................................................. 2
   The OpenStack architecture .......................................................................................... 2
   OpenStack terminology .............................................................................................. 4

3 IMPLEMENTATION SCENARIOS ..................................................................................... 5
   Cloud computing ....................................................................................................... 5

4 LAUNCHING AN INSTANCE WITH HORIZON ................................................................ 7
   Objectives .................................................................................................................. 7
   Workshops .................................................................................................................. 7
   Logging into the Horizon web interface ....................................................................... 7
   Work with self-signed certificates .............................................................................. 8
   User in a project .......................................................................................................... 9
   Components required to launch an instance .............................................................. 9
   Demonstration: Launching an instance in Horizon .................................................... 12

5 RELATED TRAINING AND CERTIFICATION ................................................................. 14

---

# LAUNCHING AN INSTANCE OF RED HAT OPENSTACK PLATFORM

WITH HORIZON (CL110)

Taste of Red Hat Training course excerpt

TECHNOLOGY DETAIL
INTRODUCTION

Taste of Red Hat Training course excerpts, labs, videos, and webinars give you a sample of the expert level of educational content you’d find in a full Red Hat® course. This excerpt is from our Red Hat OpenStack Administration I (CL110) course.

Red Hat OpenStack Administration I teaches system administrators how to implement a cloud computing environment using Red Hat OpenStack® Platform, including installation, configuration, and maintenance. Through hands-on labs, students will explore fault-tolerant and redundant configurations of Red Hat OpenStack Platform and the future plans of the OpenStack development community. This course will provide a proof of concept that can help you prepare for Red Hat OpenStack Administration II (CL210) and the Red Hat Certified System Administrator in Red Hat OpenStack exam (EX210).

In this course excerpt, you’ll find a guide to launching an instance of Red Hat OpenStack Platform with Horizon.

INDUSTRY OPPORTUNITY

THE OPENSTACK ARCHITECTURE

The OpenStack platform is implemented as a collection of interacting services that control compute, storage, and networking resources.

OpenStack services that are covered in Red Hat OpenStack Administration I:

- **Horizon (dashboard)**: A web-based interface for managing OpenStack services. It provides a graphical user interface for operations such as launching instances, managing networking, and setting access controls.

- **Keystone (identity)**: A centralized identity service that provides authentication and authorization for other services. Keystone also provides a central catalog of services running in a particular OpenStack cloud. It supports multiple forms of authentication, including user name and password credentials, token-based systems, and Amazon Web Services (AWS)-styled logins. Keystone acts as a single sign-on (SSO) authentication service for users and components. This service is responsible for creating and managing users, roles, and projects/tenants.
• Users: The Keystone service validates that incoming requests are made by legitimate users and may be assigned a token to access particular resources based on their role(s) in the project.

• Projects: A project in Keystone is equivalent to a tenant in previous versions of Red Hat OpenStack Platform. A project is a group of items (users, images, instances, network(s), volumes, etc.). It helps isolate or group identity objects. Depending on the service provider, a project (or tenant) can map to a customer, account, organization, or environment.

• Neutron (network): A software-defined networking service that helps to create networks, subnets, routers, and floating IP addresses. Administrators can create and attach interface devices managed by other OpenStack services to networks. OpenStack networking ships with plug-ins and agents for Cisco virtual and physical switches, Open vSwitch, and others. The common agents are L3 and Dynamic Host Configuration Protocol (DHCP), which provides DHCP IP addresses to instances. OpenStack networking enables projects (or tenants) to create advanced virtual network topologies, including services such as firewalls, load balancers, and virtual private networks (VPNs). The public and private networks, and floating IP address, were provided by the Neutron service.

• Cinder (block storage): A service that manages storage volumes for virtual machines. This is persistent block storage for the instances running in Nova. Snapshots can be taken for backing up data, either for restoring data or to be used to create new block storage volumes. This is often used in instances for storage such as database files.

• Nova (compute): A service that manages networks of virtual machines running on nodes, providing virtual machines on demand. Nova is a distributed component and interacts with Keystone for authentication, Glance for images, and Horizon for web interface. Nova is designed to scale horizontally on standard hardware, downloading images to launch instances as required. Nova compute uses libvirt, qemu, and kvm for the hypervisor.

• Glance (image): A service that acts as a registry for virtual machine images, allowing users to copy server images for immediate storage. These images can be used as templates when setting up new instances.

• OpenStack Networking: A service that provides connectivity between the interfaces of other OpenStack services, such as Nova. Due to OpenStack Networking’s pluggable architecture, users can create their own networks, control traffic, and connect servers to other networks. Various networking technologies are supported.

OpenStack services that are not covered in Red Hat OpenStack Administration I, but are supported by Red Hat:

• Swift (object storage): A service that provides object storage that allows users to store and retrieve files. Swift architecture is distributed to allow for horizontal scaling and to provide redundancy as failure-proofing.

• Heat (orchestration): A service to orchestrate multiple composite cloud applications using the AWS CloudFormation template format, through both a representational state transfer (REST) application programming interface (API) and a CloudFormation-compatible query API.
• **Ceilometer (telemetry):** The OpenStack telemetry service provides user-level usage data, which can be used for customer billing, system monitoring, or alerts. It can collect data from notifications sent by OpenStack services like compute usage events, or by polling OpenStack infrastructure resources. Additionally, the service provides a plug-in system that can be used to add new monitoring metrics.

• **Gnocchi (time series database):** Gnocchi is a Time-series Database-as-a-Service (TDBaaS), which is used to store metrics and metadata for the resources collected by Ceilometer into this database. It uses Swift for its storage.

• **Trove (relational database):** Trove is a Database-as-a-Service (DBaaS). Its mission is to provide scalable and reliable relational database engines. The service provides isolation at high performance while automating complex database administrative tasks, including deployment, configuration, patching, backups, restores, and monitoring.

• **Manila (shared file storage):** Manila is a secure file share as a service. It uses Network File System (NFS) and Common Internet File System (CIFS) protocols for sharing the files. It can be configured to run on a single-node back end or across multiple nodes.

• **Sahara (data processing):** Sahara aims to provide users with a simple means to provision a data processing cluster (such as Hadoop, Spark, and Storm) on OpenStack.

• **TripleO or OpenStack on OpenStack (OOO).** TripleO is used in installing, upgrading, and operating OpenStack clouds using OpenStack’s own services as the foundations. It uses Nova, Neutron, and Heat, and other orchestration tools, like Chef or Puppet, to automate fleet management, including scaling up and down at datacenter scale.

• **Ironic (bare-metal provisioning):** Ironic is an OpenStack project that provisions physical hardware as opposed to virtual machines. It provides several drivers like PXE and IPMI to cover a wide range of hardware. It also allows vendor-specific drivers to be added.

• **Tempest (integration testing):** Tempest provides a set of integration tests to be run against a live OpenStack cluster. It has several tests for OpenStack API validation, scenarios, and other specific tests useful in validating an OpenStack deployment.

**OPENSTACK TERMINOLOGY**

OpenStack uses the following terminology:

• **Cloud controller:** The coordinating manager. All machines in the OpenStack cloud communicate with the cloud controller using the Advanced Message Queuing Protocol (AMQP). In Red Hat OpenStack Platform, there are two options for AMQP: the Apache Qpid messaging daemon (qpidd) and RabbitMQ.

• **Cloud types:** Cloud computing can be deployed in three forms: public, private, and hybrid clouds. These different forms, depending on the kind of data being worked with, each provide different levels of security and management.

• **Cloud models:** There are three models under which a cloud service can be categorized based on the service it delivers: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). Red Hat OpenStack Platform is based on IaaS, which allows consumers to build their infrastructure for various deployments, whereas Red Hat OpenShift is PaaS, which offers a scalable platform to host applications.
• **Compute node:** A hypervisor – any machine running the Nova compute service. Most often, these only run the Nova compute service.

• **Volume (block storage):** A persistent disk presented to the instance. Volumes are persistent and can be attached to or detached from running instances. The Cinder service uses a logical volume manager (LVM) by default as a back end, and volumes are presented as raw devices on the instances. Logical volumes are created from this volume group. Volume snapshots can be created, similar to normal logical volume snapshots.

• **Ephemeral disk:** A temporary disk used by an instance. When the instance is created, the ephemeral disk is created as a QCOW2 image in /var/lib/nova/instances/instance-00000000X/disk.1ocal on the compute node. When the instance is terminated, this disk is removed, after being erased with dd. The first ephemeral disk normally appears as /dev/vdb within the instance.

• **Server or instance:** A virtual machine.

• **Flavor:** The hardware associated with an instance. This includes, RAM, CPUs, and disks.

• **Stack:** A group of instances built from a template. Template files are written in JavaScript Object Notation (JSON), a data exchange format designed to be a simpler alternative to Extensible Markup Language (XML). Document encoding stacks and the template files are used in the Heat orchestration service.

• **OpenStack Networking (Neutron):** The OpenStack networking API uses these abstractions to describe network resources:

  • **Network:** An isolated L2 segment, analogous to a virtual local area network (VLAN) in the physical networking world.

  • **Subnet:** A block of IPv4 or IPv6 addresses and associated configuration state.

  • **Port:** A connection point for attaching a single device, such as the network interface card (NIC) of a virtual server, to a virtual network. Also describes the associated network configuration, such as the media access control (MAC) and IP addresses to be used on that port.

  • **Open vSwitch:** Software that provides a virtual switch. Open vSwitch provides traffic queuing and shaping and automated flow control. The Open vSwitch plug-in will be used for OpenStack networking.

**IMPLEMENTATION SCENARIOS**

**CLOUD COMPUTING**

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. These computing resources can be networks, servers, storage, applications, or services. A user is able to launch as many instances (virtual machines) as needed, without requiring provision assistance from the service provider. The cloud consumer decides which computing resources to use for their instances. Cloud computing has several essential characteristics:

• **Self-service:** Allows cloud consumers to provision instances with computing resources.

• **Multitenancy:** Allows multiple cloud consumers to share the underlying hardware.

• **Elasticity:** Scales out (or scales in) instances to satisfy demand.
• **Telemetry:** Resources can be monitored and metered by the service provider and the cloud consumer.

**Traditional workloads vs. cloud workloads**

Traditional workloads or datacenter virtualization has been common in the computer industry for many years. Traditional workloads use a client-server architecture; failover and scaling are built into the infrastructure. One configurable machine is built to handle the workload. When the workload increases, the machine scales up by adding more RAM, more CPUs, and more storage.

Cloud workloads require design changes to the application. The application uses a distributed architecture. Failover and scaling are built into the application. The application can scale out by adding more virtual instances to meet demand.

**Benefits of OpenStack**

Red Hat OpenStack Platform permits heterogeneous environments supporting cloud deployments to integrate with OpenStack technologies—which are based on open standards—all with the flexibility for future changes. As innovation is made within the OpenStack community and released by the OpenStack Foundation, Red Hat will ensure its platform maintains full interoperability for ongoing integration of all cloud infrastructure components in an environment. Perpetual interoperability coupled with the broad vendor ecosystem ensures companies that adopt Red Hat OpenStack Platform can continue to purchase low-cost commodity hardware to meet demand and replace end-of-life equipment. With the Red Hat platform, customers are not subject to cost and availability pitfalls that can happen with a proprietary approach. Some of the benefits of using OpenStack for building private clouds are:

• **Standardization in its foundation.** The promise of the cloud has greatly eliminated the days when large technology providers tried to one-up each other with giant, closed systems. But that promise is dependent on standardization. Thus, more than 200 companies supporting OpenStack are striving toward a flexible, standardized platform that works interchangeably with any infrastructure. This is extremely important, especially since many financial companies have spent years investing millions of dollars in IT.

• **Less cost and more innovation.** Most IT departments are focused on running and managing the infrastructure and not providing innovative solutions. The flexibility and low cost of OpenStack helps alleviate this by freeing up IT to focus on new applications, solutions, and service delivery rather than an inflexible underlying infrastructure. This allows for faster delivery of new features and products, such as online tools to help customers better manage their portfolios, and can help attract customers and increase retention.

• **Industry-wide support.** OpenStack receives widespread support from some of the most important players in the technology industry, all of which have come together to help companies break away from being locked in with a particular cloud vendor. While some of these players offer their own flavor of OpenStack, they still commit to the ideals of an open, standardized cloud. Therefore, unlike many technology purchases, it is not really about choosing the technology itself, but selecting a vendor with the richest ecosystem and support, knowing that support extends to virtually an entire industry.
• **Portability to other clouds.** Investments in open cloud development like OpenStack must be portable to other clouds. Portability takes a variety of forms, including programming languages and frameworks, data, and applications. If developing an application for one cloud, it should not need to be rewritten in a different language or use different APIs to move it somewhere else.

• **Deployable on the infrastructure of choice.** OpenStack provides an additional layer of abstraction above virtualization, physical servers, storage, networking, and public cloud providers. This requires that cloud management be independent of specific virtualization and other infrastructure technologies. OpenStack can work on libvirt, kvm, or qemu.

**LAUNCHING AN INSTANCE WITH HORIZON**

**OBJECTIVES**

• Log into Horizon.

• Work with self-signed certificates.

• Describe a project and its associated users.

• Launch an instance in Horizon.

• Verify the instance launched.

Note: This course excerpt is just the beginning of a full, deep dive into administrating Red Hat OpenStack Platform. Further chapters in this course show you how to:

• Deploy each Red Hat OpenStack Platform service manually.

• Manage users and projects.

• Deploy instances and use Heat to deploy and customize instances.

**WORKSHOPS**

Logging into the Horizon web interface

The OpenStack dashboard is a web-based graphical user interface for managing OpenStack services. This dashboard is also known as Horizon. It allows customizing the brand of the dashboard.
Accessing the browser dashboard requires the host name and login password. It is accessible over http or https, for example: https://servera.lab.example.com/dashboard.

**Work with self-signed certificates**

Normally, a certificate authority (CA) would be used to generate the certificates used by OpenStack, including the web server certificate for Horizon. In the classroom, the default certificate mechanisms will be used, which will generate a self-signed certificate. Firefox does not accept these self-signed certificates by default because of the security implications. To manually accept the certificate, follow these steps.

1. Browse to the Horizon URL.
2. Expand the I Understand the Risks menu.
3. Select the Add Exception... option.
4. Select the Confirm Security Exception option to permanently store this exception.

If OpenStack has to be reinstalled, the self-signed certificate will remain in the browser, and when trying to connect to the new OpenStack installation, an error will appear:

**Secure Connection Failed**

An error occurred during a connection to demo.example.com. You have received an invalid certificate.

Please contact the server administrator or email correspondent and give them the following information:

Your certificate contains the same serial number as another certificate issued by the certificate authority.

Please get a new certificate containing a unique serial number.

(Error code: sec_error_reused_issuer_and_serial)

This error states that the serial number has been reused by the certificate authority. To remove the old certificate and certificate authority in Firefox, follow these steps:

1. Open the Firefox menu by pressing the Alt key.
2. Go to Edit > Preferences.
3. Go to the Advanced icon and select the Certificates tab.
4. Select the View Certificates option.
5. In the Authorities tab, scroll down to the openstack CA (if it exists). Highlight the server name and select the Delete or Distrust... option. Select OK to confirm. Do the same to any other CAs listed under the openstack list.
6. Go to the Servers tab. In the openstack accordion, highlight the server name and select the Delete option. Press OK to confirm. Repeat for any other servers listed under openstack.
7. Close the Preferences windows.
8. Open the menu again by pressing Alt. Go to History > Clear Recent History....
9. In **Time range to clear**, choose **Everything**. Select **Cache** and **Active Logins** check boxes. Select the **Clear Now** option.

10. Go to the **Horizon** dashboard URL and accept the self-signed certificate as outlined previously.

**User in a project**

The **Identity** tab provides an interface for viewing and managing projects and users.

To make an **OpenStack** cloud, **multitenant projects** are used to act as a container of resources owned by different users.

A set of resource quotas are preconfigured when a new project is created. The quotas include the number of instances, number of VCPUs, RAM, and floating IPs that can be assigned to instances in the project. Users can be assigned multiple projects, but one of the projects is designated the **primary project**. The **primary project** is simply the first project the user is associated with.

The **Horizon** dashboard or command-line tools can be used to create, modify, and delete projects, view project usage, and add or remove users as project members, modify quotas, and set an active project.

Using the **Horizon** dashboard or command-line tools, users with the admin role associated in the admin project can view, create, edit, and delete users, and change user passwords. The **Users** tab displays only if logged in as a user with administrative privileges. While creating users, the **Role** needs to be specified apart from username, password, email address, and primary project. **OpenStack** comes with two predefined roles, **_member_**, and **admin**. Adding a user with **admin** role makes the user an administrative super user.

Components required to launch an instance

To launch an instance in **OpenStack**, the following components must be set up first:

**Images.** An image is a file that contains a virtual disk with a bootable operating system installed on it. Images can either be created or customized using **openstack image create**. Several prebaked images provided by various software vendors can also be imported. In a multitenant cloud environment, users can also share their personal images with other projects. These images can be of various formats (RAW, QCOW2, ISO, VMDK, VHD, ARI, AKI and AMI), which can be imported into OpenStack.
Images for the current project can be found in the Horizon dashboard under Images. There is a Create Image option on the main Images page for creating new images. Flavors. Virtual hardware templates are called flavors in OpenStack. It defines sizes for minimum RAM and disk and image format. The Horizon dashboard and command-line tools provide the ability to modify and delete an existing flavor and create a new one.

Flavors for the current project can be found in the Horizon dashboard while launching an instance by a minimal user, but only administrators have permission to edit or create new flavors. (The rights may also be delegated to other users.) An admin user can use the Admin tab, which provides a Flavors item.

Security groups. Security groups are sets of IP filter rules that are applied to an instance's networking. They are project-specific, and project members can edit the default rules and add new rule sets. All projects have the default security group pre-created, which is applied to instances that have no other security group defined. Unless edited, the default security group denies all incoming traffic.

Security groups for the current project can be found in the Horizon dashboard under Access & Security. There is a Create Security Group option on the main Access & Security page for creating new security groups.
Key pair. To improve the security posture in the cloud, logging into the cloud instances with secure shell (SSH) is done by use of a public and private key pair instead of using a username and password. These key pairs can be created or imported via the Horizon dashboard under Access & Security. There is a Create Security Group option on the main Access & Security page for creating or importing key pairs.

Networks. Networks define the network routes to be added to an instance. Based on the network the instance belongs to and the route table for the network, the instance can be public-facing or private-facing.

A floating IP address can be associated with the private-facing interface to make it public-facing and potentially reachable outside the OpenStack network. Administrators can configure rich network topologies by creating and configuring networks and subnets, and then instructing other OpenStack services to attach virtual devices to ports on these networks.

OpenStack networking supports multiple private networks for each project and allows projects to choose their own IP addressing scheme, even if IP addresses overlap with those used by others.

The network topology can be viewed via the Horizon dashboard under Project by selecting the Network Topology menu item under the Network menu.
**Launch an instance in Horizon.** Horizon can be used to launch and manage instances after they have spawned. Before launching an instance, it will be necessary to create and upload an image (which constitutes the operating system and software for the instance), configure a security group (to open certain ports to the instance through the firewall), create an SSH key pair (to be able to connect to the instance), and associate a floating IP address (to be able to access the instance from outside). The following demonstration will walk through the steps of launching the first instance.

**Demonstration: Launching an instance in Horizon**

2. Log into the Horizon dashboard using **demo_user** as the username and the password as **redhat**.
3. Click the Project dropdown to verify the demo_project is marked as current.
4. Launch an instance named **demo_instance** using **m1.small** flavor, the **demo_image** image, the **demo_instance-key** key pair, the **demo_instance-sg** security group, and the private network.
   - Go to the **Instances** subtab under the **Compute** tab.
   - Select the **Launch Instance** option.
   - In the **Details** tab, enter **demo_instance** as the **Instance Name**.
   - In the **Details** tab, choose **m1.small** as the **Flavor**.
   - In the **Details** tab, choose **Boot from image** in the **Instance Boot Source** dropdown menu.
   - In the **Details** tab, select **demo_image** as the **Image Name**.
   - In the **Access & Security** tab, make sure the **demo_instance-key** key pair is selected. Select the **demo_instance-sg** security group.
   - In the **Networking** tab, ensure that **private** network is selected. If its not selected, click the + button next to the **private** (private) network to select it. Select the **Launch** option to launch the **demo_instance**.
5. To associate a floating IP address 172.25.250.Y to the demo_instance instance:
   • Inside the Instance subtab in demo_instance row, open the dropdown menu under Actions.
   • Select the Associate Floating IP menu item.
   • Choose 172.25.250.Y as the floating IP address from the IP Address dropdown, and choose demo_instance: 192.168.2.2 under Port to be associated. Replace the IP address with the one that has been associated to the instance demo_instance.
   • Select the Associate option to associate the IP address with demo_instance.

6. Right-click the demo_instance instance link and choose Open Link in New Tab. In the new tab, choose the Console tab, then select the Click here to show only console link. If a certificate error appears, accept the self-signed certificate. Watch the virtual machine boot. (It may have already booted.)

7. To verify the setup, open a new terminal on workstation and SSH to 172.25.250.Y. Use the private key stored as /home/student/Downloads/demo_instance-key.pem on workstation.

8. [student@workstation ~]$ ssh -i ~/Downloads/demo_instance-key.pem root@172.25.250.Y
   The authenticity of host ‘172.25.250.Y (172.25.250.Y)’ can’t be established.
   Are you sure you want to continue connecting (yes/no)? yes
   Warning: Permanently added ‘172.25.250.Y’ (RSA) to the list of known hosts.
   Please login as the user “cloud-user” rather than the user “root”.
   Connection to 172.25.250.Y closed.
   [student@workstation ~]$ ssh -i ~/Downloads/demo_instance-key.pem cloud-user@172.25.250.Y
   [cloud-user@demo-instance ~]$ exit

9. The ping command is another common network test, but the demo_instance-sg security group did not allow ICMP. Try to ping the demo_instance from workstation.

10. [student@workstation ~]$ ping -c3 172.25.250.Y
    PING 172.25.250.Y (172.25.250.Y) 56(84) bytes of data.
    --- 172.25.250.Y ping statistics ---
    3 packets transmitted, 0 received, 100% packet loss, time 2000ms
   • Back in the Horizon dashboard, while still logged in as the demo_user user, go to the Compute tab.
   • Choose the Access & Security subtab.
   • In the Security Groups tab, select the Manage Rules option in the demo_instance-sg row.
   • Select the + Add Rule option. In the Rule menu, choose ALL ICMP, add select the Add option.
   • Try to ping the demo_instance instance again.
TECHNOLOGY DETAIL  Launching an instance of Red Hat OpenStack Platform with Horizon (CL110)

[student@workstation ~]$ ping -c3 172.25.250.Y
64 bytes from 172.25.250.Y: icmp_seq=1 ttl=63 time=0.642 ms
64 bytes from 172.25.250.Y: icmp_seq=2 ttl=63 time=0.457 ms
64 bytes from 172.25.250.Y: icmp_seq=3 ttl=63 time=0.596 ms

--- 172.25.250.Y ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.457/0.565/0.642/0.078 ms

RELATED TRAINING AND CERTIFICATION
Want to learn more about how to install, configure, and maintain Red Hat OpenStack Platform? Our hands-on, lab-based course can teach you much more. Then, prove your skills by achieving your certification.

Course
Red Hat OpenStack Administration I (CL110)
Red Hat OpenStack Administration II (CL210)
Red Hat OpenStack Administration III (CL310)

Certification
Red Hat Certified System Administrator in Red Hat OpenStack
Red Hat Certified Engineer in Red Hat OpenStack

About the Authors
Adolfo Vazquez
Michael Jarrett
Razique Mahroua
Snehangshu Karmakar

ABOUT RED HAT
Red Hat is the world's leading provider of open source software solutions, using a community-powered approach to provide reliable and high-performing cloud, Linux, middleware, storage, and virtualization technologies. Red Hat also offers award-winning support, training, and consulting services. As a connective hub in a global network of enterprises, partners, and open source communities, Red Hat helps create relevant, innovative technologies that liberate resources for growth and prepare customers for the future of IT.